

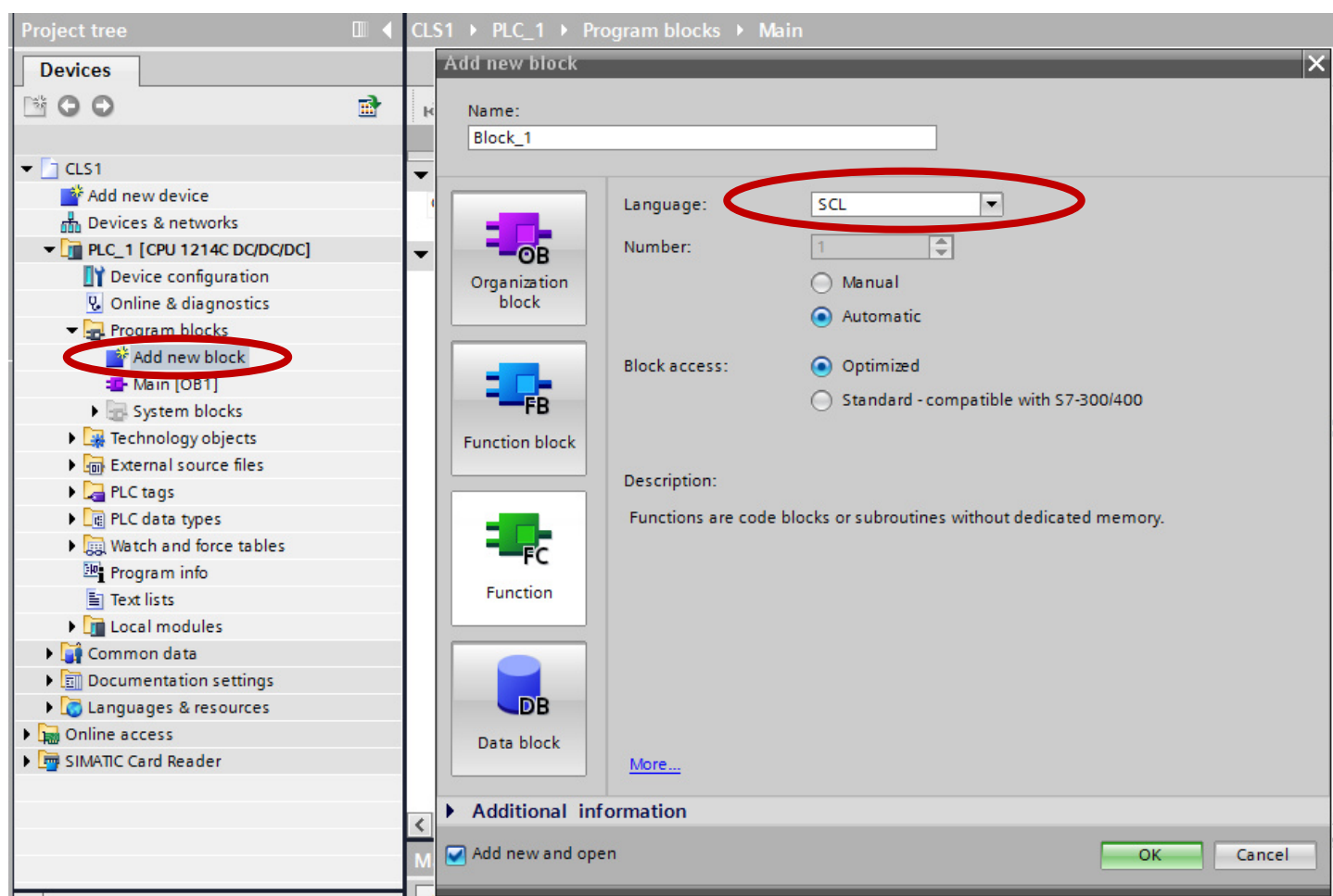
Podstawy programowania w SCL.

Co to jest SCL:

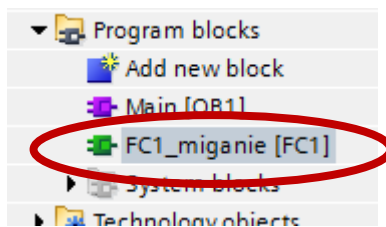
SCL (ang. Structured Control Language) to język wysokiego poziomu programowania bazujący na języku **PASCAL**. Ujęty on został w normie **PN-EN 61131-3** (międzynarodowej IEC 1131-3) przewidzianej do programowania sterowników programowalnych. Język programowania SCL opracowany przez firmę Siemens spełnia poziom podstawy PLCopen języka **ST** (ang. Structured Text) określonego w powyższej normie.

Jak używać języka SCL:

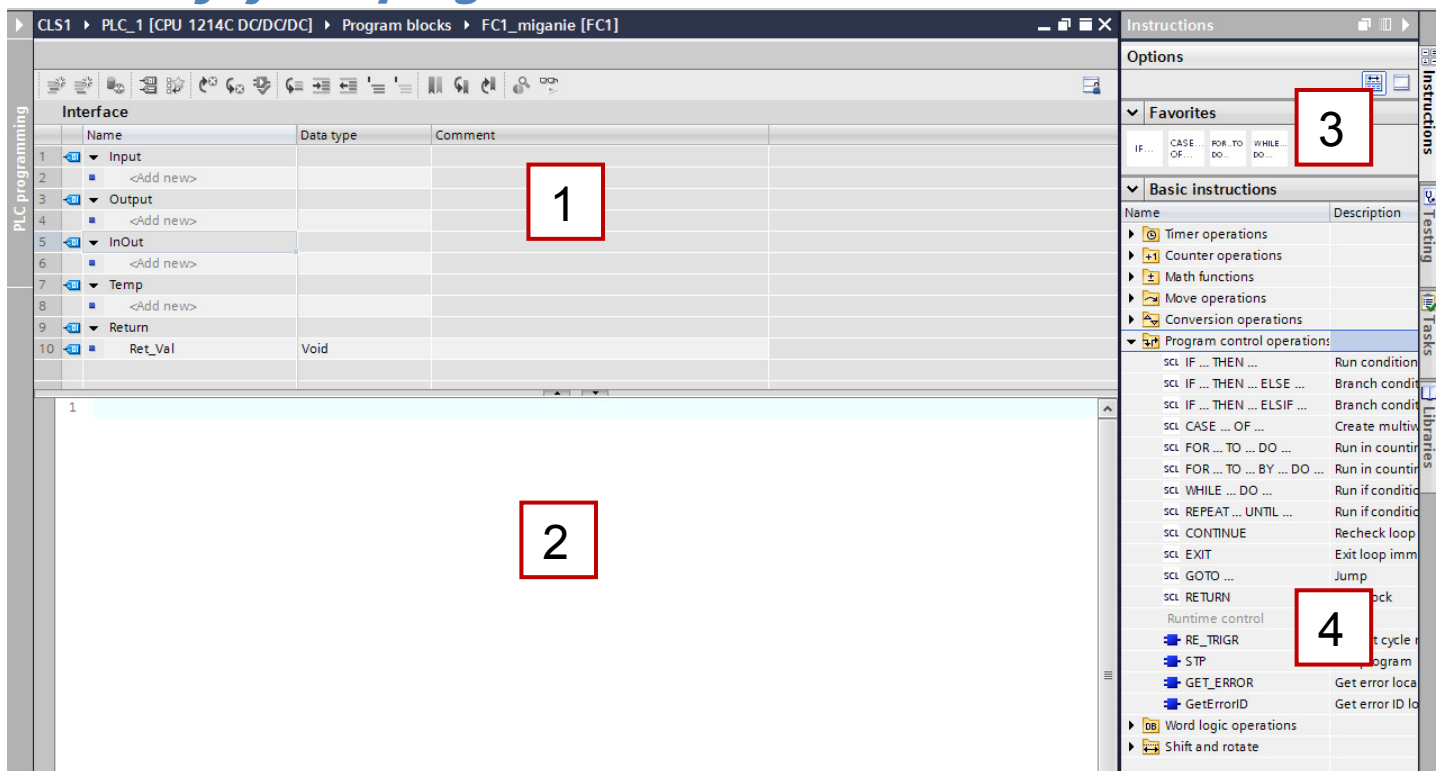
Po założeniu nowego projektu i przygotowaniu TIA do programowania możemy używać języka SCL w nowych blokach lub funkcjach:



Po wybraniu rodzaju obiektu programowego nadaj mu nazwę i wybierz język programowania (SCL). Po tej operacji pojawi się w drzewie projektu nowy obiekt:



Okno edycji dla programowania w SCL:



1. Okno ustawień danych wejściowych i wyjściowych dla funkcji (są to zmienne lokalne i będą widoczne tylko wewnątrz funkcji)
2. Okno edycji kodu SCL.
3. Szybki dostęp do instrukcji SCL.
4. Biblioteka wszystkich dostępnych instrukcji językowych.

Najważniejsze elementy języka SCL:

- **ograniczniki** (ang. delimiters) to znaki specjalne stosowane do różnych celów, np. +, -, %, #,), (, *, \$;
- **słowa kluczowe** (ang. keywords) to nazwy (identyfikatory danego języka) stosowane w danym języku i nie wolno ich używać jako nazw własnych przez użytkownika, np. **FUNCTION, BOOL, VAR_INPUT, TIME, OR, MOV, TIM, I, OUT;**
- literały (ang. literals) do zapisywania danych (stałych np. **-15, TIME#15h_12m, #50** lub zmiennych np. **VAR_INPUT**

X:INT; END_VAR) w postaci numerycznej, ciągu znaków lub czasu;

- **identyfikatory** (ang. identifiers) to nazwy własne programisty w postaci ciągów znaków alfanumerycznych, np. **styk5** lub **STYK_5**;
- **komentarze**.

Operatory języka SCL:

Typ operatorów	Typ operacji	Operator
Nawiasy	Kolejność wykonywania (nawiasy okrągłe)	()
Arytmetyczne	Potęgowanie	**
	Jednoargumentowy plus (znak)	+
	Jednoargumentowy minus (znak)	-
	Mnożenie	*
	Dzielenie	/
	Funkcja modulo	MOD
	Dodawanie	+
	Odejmowanie	-
Porównania	Mniejszy niż	<
	Mniejszy niż lub równy	<=
	Większy niż	>
	Większy niż lub równy	>=
	Równy	=
	Różny (nierówny)	<>
Logiczne (binarne)	Negacja	NOT
	Iloczyn logiczny, I	AND lub &
	Alternatywa wykluczająca	XOR
	Alternatywa, LUB	OR
Przypisania	Przypisanie	:=

Instrukcje:

Favorites	
IF...	CASE... OF...
FOR... TO DO...	WHILE... DO...
(*...*)	
Basic instructions	
Name	
▶ Bit logic operations	IF...THEN... instrukcja wykonania warunku
▶ Timer operations	IF...THEN...ELSE... instrukcja wykonania kilku warunków
▶ Counter operations	IFTHEN.ELSIF... instrukcja wykonania kilku warunków wielokrotnie
▶ Math functions	CASE...OF... instrukcja wyboru
▶ Move operations	FOR...TO...DO... instrukcja pętli
▶ Conversion operations	FOR...TO...BY...DO... instrukcja pętli z wartością skoku
▼ Program control operations	WHILE...DO... instrukcja pętli, jeżeli warunek spełniony
scl IF ... THEN ...	REPEAT...UNTIL... instrukcja pętli, jeżeli warunek niespełniony
scl IF ... THEN ... ELSE ...	CONTINUE natychmiastowe przejście do kolejnej iteracji
scl IF ... THEN ... ELSIF ...	EXIT opuszczenie natychmiastowe pętli
scl CASE ... OF ...	GOTO skok
scl FOR ... TO ... DO ...	RETURN opuszczenie bloku
scl FOR ... TO ... BY ... DO ...	Wprowadzenie komentarzy do bloku
scl WHILE ... DO ...	RE_TRIGR restart czasu monitorowania cyklu
scl REPEAT ... UNTIL ...	STP opuszczenie programu
scl CONTINUE	GET_ERROR wywołanie błędu lokalnego
scl EXIT	GET_ERR_ID wywołanie ID błędu lokalnego
scl GOTO ...	INIT_RD inicjalizacja przechowywanych danych (tylko w S7-1500)
scl RETURN	WAIT opóźnienie wykonania programu w ms (tylko w S7-1500)
scl Insert block comment	RUNTIME pomiar czasu pracy programu (tylko w S7-1500)
Runtime control	
RE_TRIGR	
STP	
GET_ERROR	
GET_ERR_ID	
INIT_RD	
WAIT	
RUNTIME	
▶ Word logic operations	
▶ Shift and rotate	
▶ ETC Additional instructions	